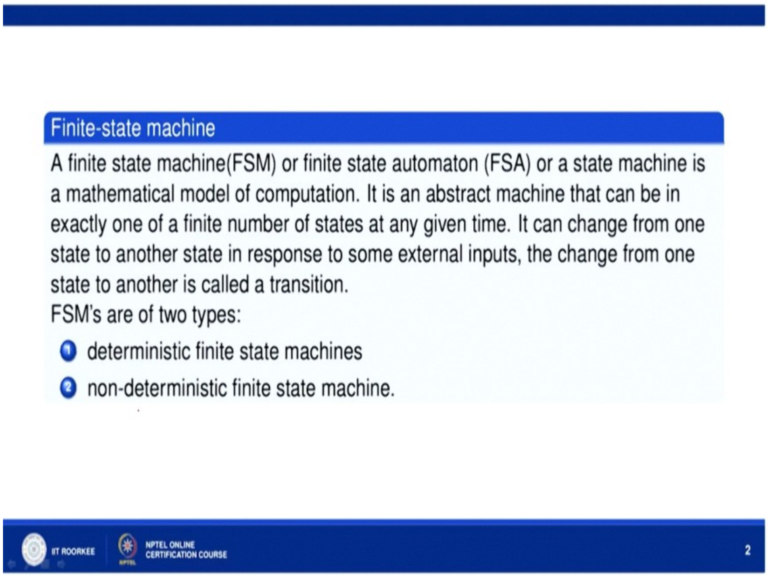
Higher Engineering Mathematics
Prof. P.N. Agrawal
Department of Mathematics
Indian Institute of Technology Roorkee
Finite – States Machines
Mod02_Lec10

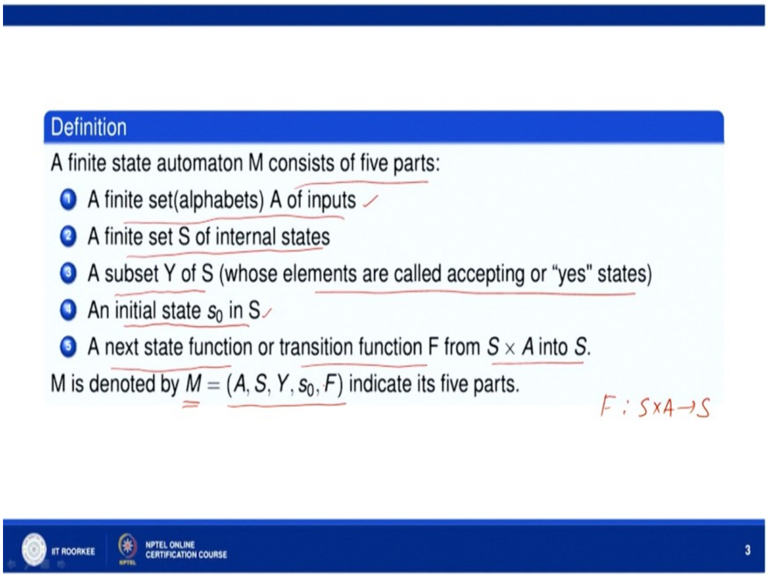
(Refer Slide Time: 0:28)



Hello friends, welcome to my lecture on Finite States Machines. A1 finite state machine we can briefly, briefly we can say FSM or Finite State Automaton which we can write FSA or a state machine is a mathematical model of computation. It is an abstract machine okay, it is an abstract machine.

A finite state machine FSM or finite state automaton FSA or a state machine is a mathematical model of computation. It is an abstract machine that can be in exactly one of a finite number of states at any given time. It can change from one state to another state in response to some external inputs, the change from one state to another is called a transition. Now, FSMs are of two types, deterministic finite state machines and nondeterministic finite state machines, we will be discussing deterministic finite state machines.

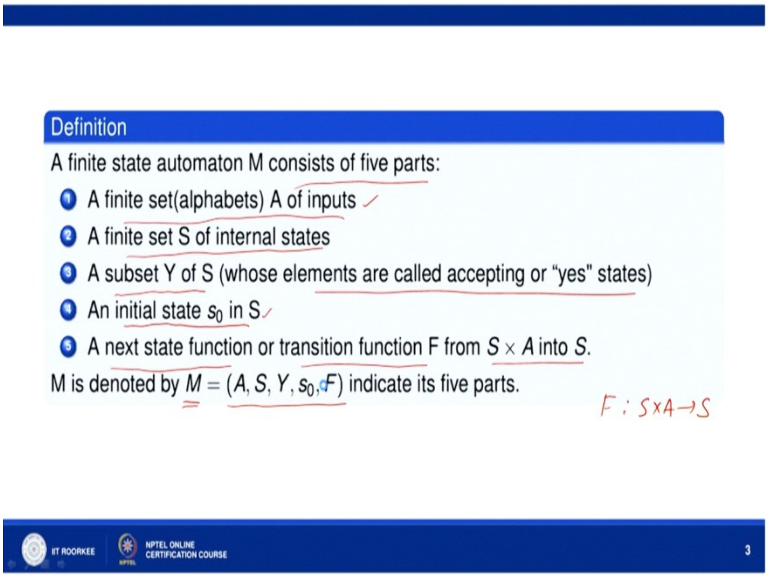
(Refer Slide Time: 1:30)

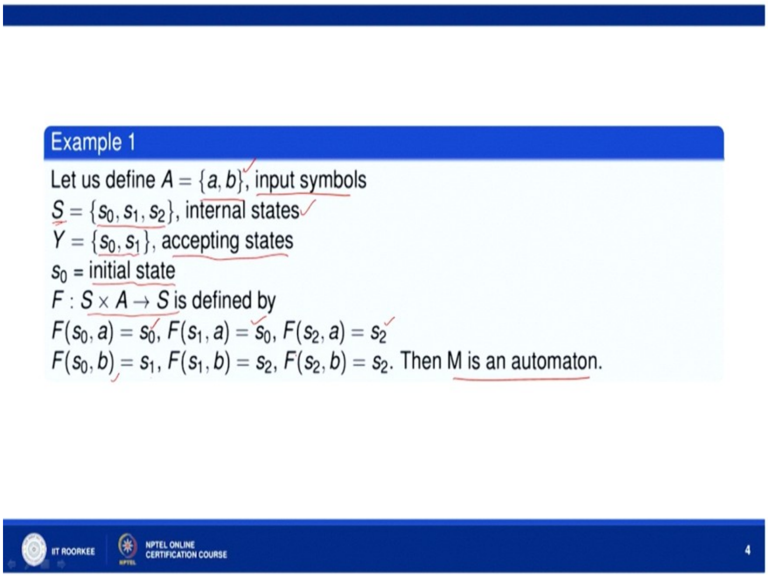


A finite state automaton M, we denoted by M. It consist of five parts, a finite set alphabets A of inputs, a finite set S of internal states, a ⊆ Y of S, whose elements are called accepting or 'yes' states and an initial state S_0 in S, a next state function or transition function F :S×A→S , so we have a transition function F $: S \times A \to S$ here, so again, let see a finite set A of inputs okay, A consist of finite number of alphabets, we have set S of internal states, it is again a finite set.

Y ⊆S, elements of Y are called accepting or yes states, an initial state S ∉S. Okay, a next state function or transition function F :S×A→S , so there are five parts. Okay and this finite state automaton M is then denoted by these five parts A, S, Y, S_0 and F. Okay, so it consist of these five parts.

(Refer Slide Time: 2:54)



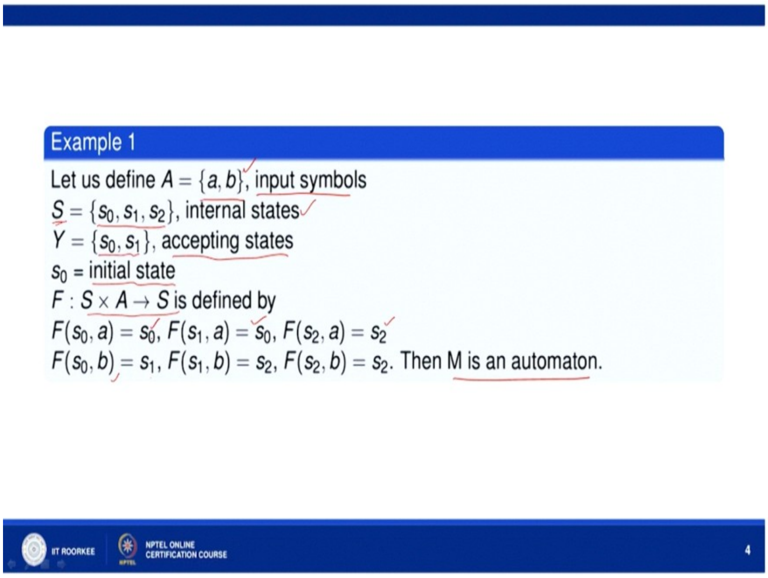


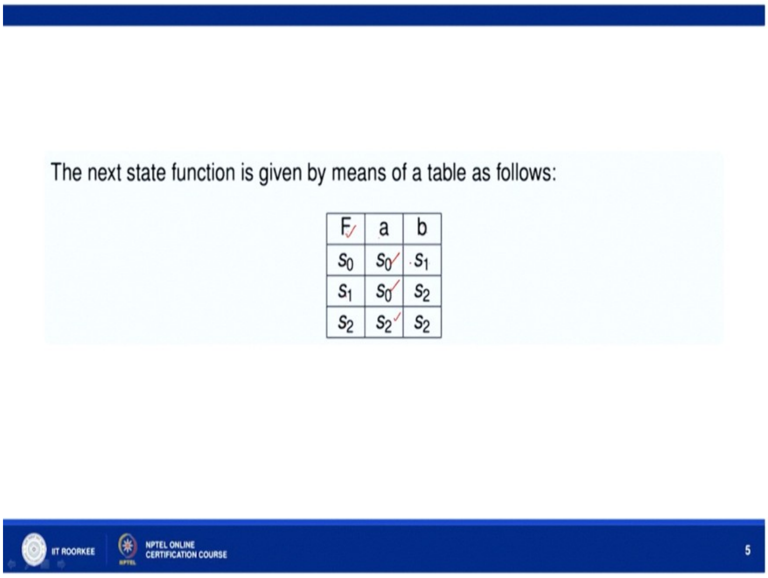
Now that is define A, A is the set of as we have said that, A is the set of alphabets that is the inputs, so let us say A = {a,b} input symbols okay, S is the internal states, set of finite number of internal states S_0, S_1, S_2. Y is a ⊆ of S and Y the element of Y are called the accepting states or yes states, S_0 is an initial state and it is a member of S, F :S×A→S and let us denote this transition function F, define this transition function F :S×A→S by means of these equations.

F¿ ¿,a) = S_0, F¿ ¿,a) =S_0, F¿ ¿,a)=S_2, F¿ ¿,b)=S_1, F¿ ¿,b) =S_2 and F¿ ¿,b)=S_2, then clearly M is an automaton because we need five things for M to be an automaton, the first thing is that it should have a finite number of input symbols, so there are finite number of input symbols here, a finite number of internal states which we have S_0, S_1, so finite number of internal states are there.

Then ⊆ of Y of S, consisting of accepting states, so S_0, and S_1 are accepting states, then we should also have S_0as initial state of S okay, so S should belong, $S_0 \in$ S, so that is there, $S_0$is already there in S and then we should be having a transition function F : S $\times A \rightarrow S$ that is also given to S, so then M must be an automaton.

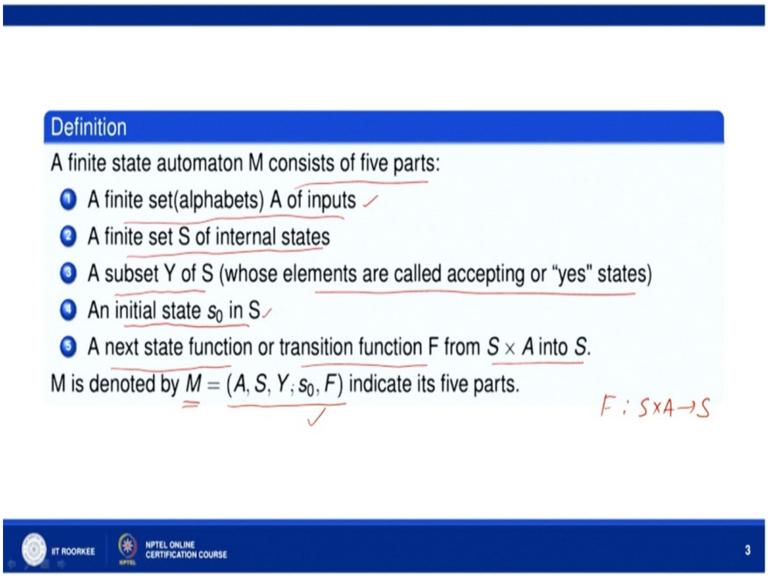
(Refer Slide Time: 4:52)

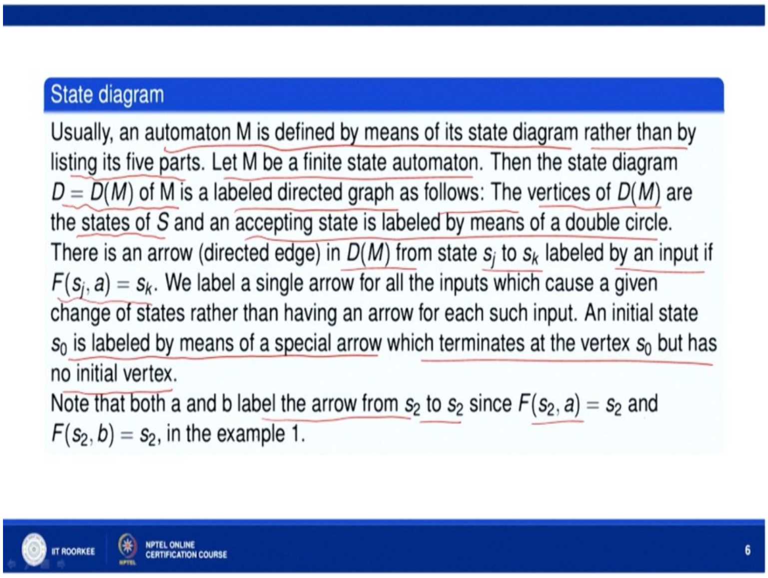




Now, the next state function can be represented by the means of this table, you can see this next state function, next state function or transition function, so F(s_0 ,a) =s_0, you can see this transition function F, so F (s_0 ,a) okay, F acting on ¿¿, a) = s_0, F (s_1 ,a) = s_0 , F (s_2 ,a) = $s_2$and then F (s_0 ,b) = s_1 okay, F (s_1 ,b) = S_2 and F (s_2 ,b) = S_2, so these six equalities okay, 1, 2, 3, 4, 5, 6, they can be represented by means of this table.

(Refer Slide Time: 5:45)





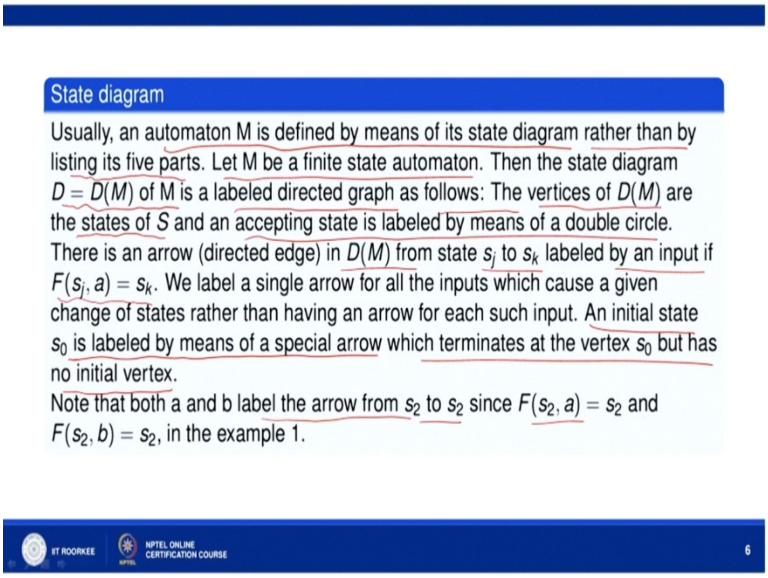
Now, let us look at state diagram or transition diagram or transition table, usually an automaton M is defined by means of its state diagram rather than by listing its five parts as we have seen that M is represented by listing its five parts ASY S_0 F, so rather than M=(A,S,Y,S_0,F) we write its state diagram okay, so it is defined by the means of state diagram.

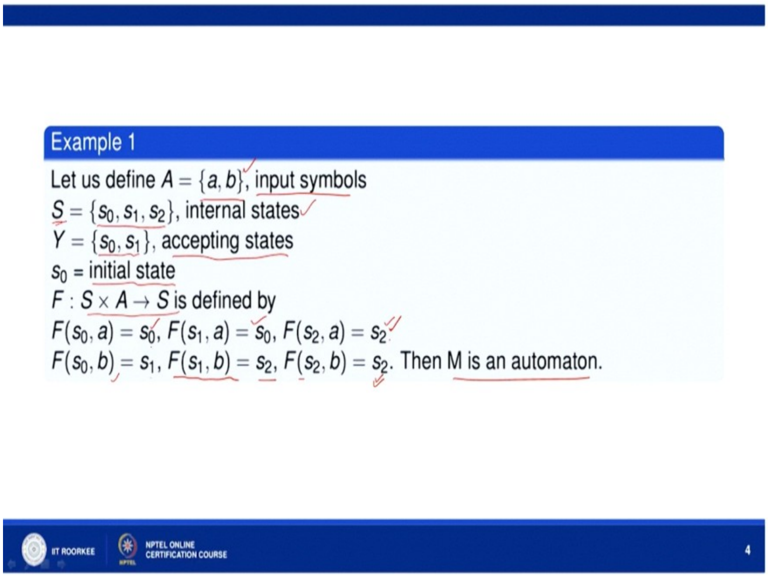
Let M be a finite state automaton okay, then the state diagram D =D(M) of the automaton M is a labeled directed graph as follows, the vertices of D(M) are the states of M, now the states of S are what? The states of S are S_0 S_1okay, so they are the vertices of D(M) okay, and an accepting state is labelled by means of a double circle okay, accepting states are what? The accepting states are $S_0$and S_1, so they are represented by means of a double circle.

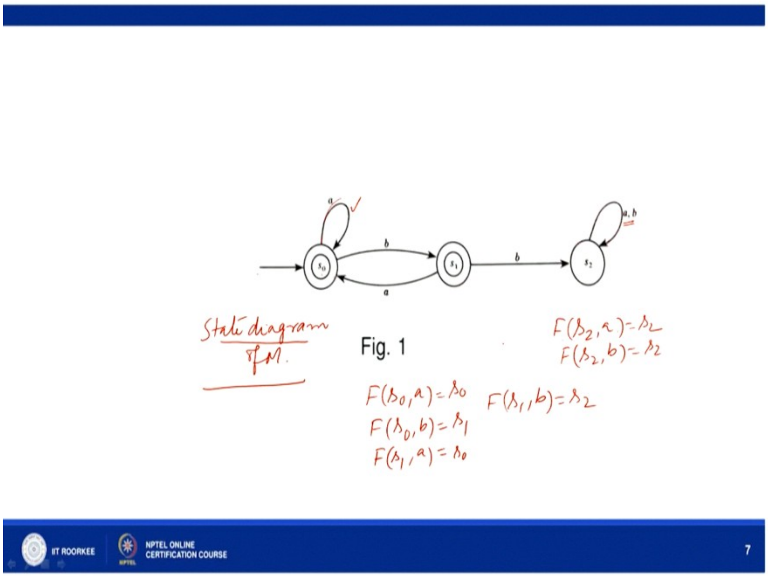
There is a narrow directed edge in D(M) from state S_j to $S_k$labeled by an input f(S_j ,a) =S_k okay, we label a single arrow for all the inputs which cause a given change of states rather than having an arrow for each such input. An initial state S_0 is labeled means of, now S_0 is labeled by means of a special arrow which terminates at the vertex S_0 but has no initial vertex okay, so we have to note this.

Now, you can see here in the transition function F ¿¿ b) =S_2, F no, F (S_2 a) =S_2, F ($S_{2,}$b) =S_2, so this one F ¿¿a) =S_2 and F(S_2,b) =S_2 both a and b label the arrow from S_2 to S_2 because F($S_{2,}$ a) =S_2 and F ¿ b) =S_2 in the example 1, in this example.

(Refer Slide Time: 8:11)







Now let us see what is D(M) ? You can see here, this as we said that an initial state S_0 is labeled by means of a special arrow, so this is a arrow which terminates at S_0 . Okay, we will begin with this arrow which terminates at S_0 because S_0 is the initial state. Now accepting states as we know, accepting states are S_0 an S_1, so they must be double circles, they must be represented by means of double circles, so you can see here S_0 is represented by means of a double circle okay and S_1 is also represented by means of a double circle okay.

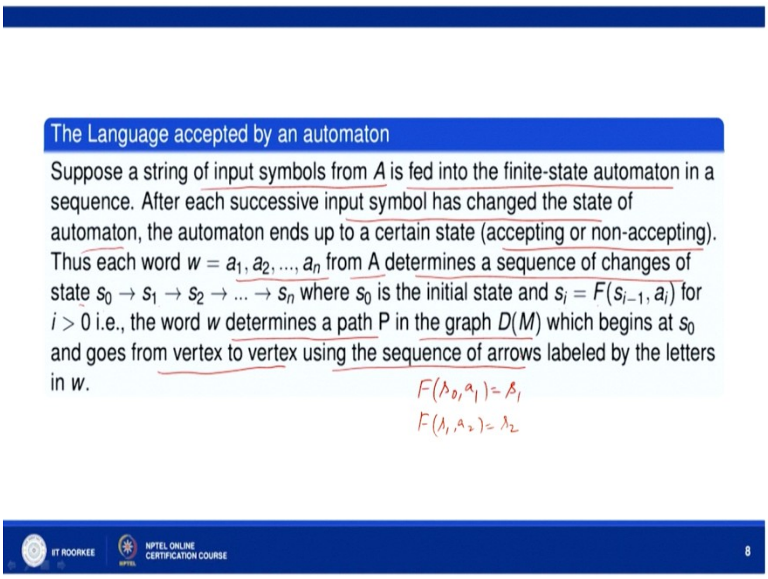
Now, let see we label a single arrow for all the inputs which cause a given change of states rather than having an arrow for each such input, you can see here S_2, F ¿¿ ,a) we have seen that F ¿ a) =S_2, F (S_2,b) =S_2 okay, so rather than saying F (S_2, a) =S_2 like that, I mean instead of saying that F ¿¿, a) =S_2 and another one, F (S_2,b) =S_2 rather than drawing two arrows, we just draw a single arrow and level it by A, B okay.

So then this single arrow where it is labeled by A, B means that F¿¿ a) =S_2 and F¿¿ b) =S_2 okay. Now yes and we also have F¿,a) =S_0. Okay, you can see here, F¿¿,a) =S_0. Okay, so ¿¿ a), F¿,a) =S_0 is represented by means of this arrow okay, which terminates at S_0, F ¿¿ a) =S_0 , now we have F¿b) =S_1 okay, so that is shown by means of this arrow F ¿¿b) =S_1 and we have F (S_1,a) =S_0, F(S_1,a) =S_0, that is shown by means of this arrow okay.

So this arrow shows F¿,a) =S_0. Okay, this arrow shows F¿b) =S_1, this arrow shows F ¿¿A) = S_0, now F¿¿,B) =S_2, so this arrow shows F¿¿,b) =S_2 and this arrow shows F¿¿, a)=S_2, F ¿¿,b) =S_2, so this is state diagram of the finite automaton M, so rather than listing its five parts state diagram is shown by its, rather than listing the all five parts of the finite state automaton M we draw its state diagram.

It clearly shows what days finite state automaton M, what are the accepting states, we can easily see from here, the internal states are what? We also can see what are the accepting states? We can also see the transition function, we can also see that S_0 , S_1 and S_2 okay, they ∈ S, the set of internal state, so all is clear from here.

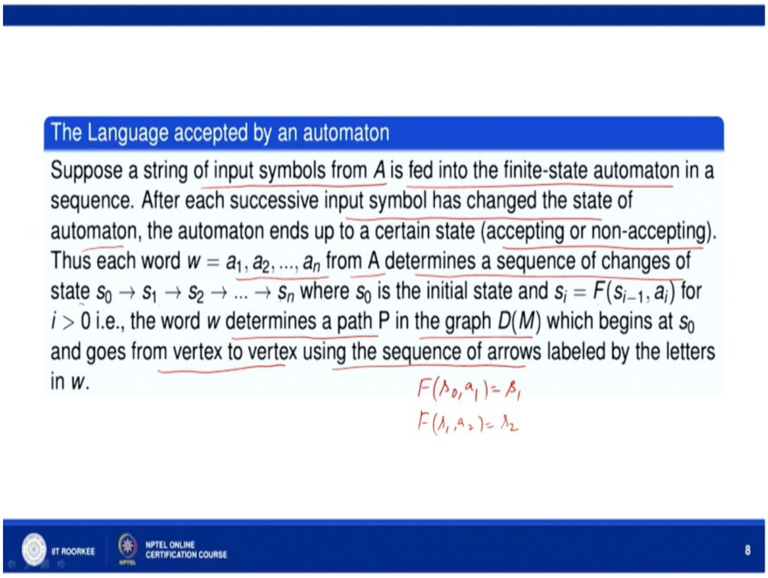
(Refer Slide Time: 12:13)

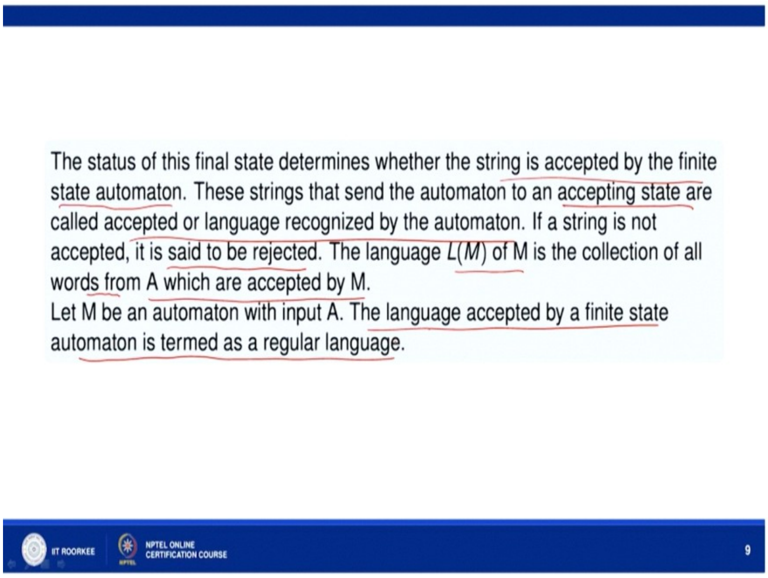


Now, let see the language accepted by an automaton, suppose a string of input symbols from A is fed into the finite state automaton in a sequence okay, after each successive input symbol has changed the state of the automaton, the automaton ends up to a certain state accepting or non-accepting, that means that state either is an element of Y or it iS_0 an element of Y Y, Y is a ⊆ of S, here it is input symbols from A okay.

So A is the set of input symbols, now thus each word W =a_1, a_2, …..a_n from A okay, if you take word W =a_1, a_2, …..a_n from A, it determines a sequence of changes of state S_0 to S_1, S_1 to S_2 and so on to S_n, where S_0 is initial state and what are S? S_i=¿ F(S_{i-1},a_i) okay, so for i > 0, i.e., what we want to say is that F ¿,a_1) =S_1, F(S_1,a_2¿ =S_2 okay, this F is the transition function okay, this word W determine is a sequence of changes of state, the word W determine is a path P in the graph D(M) which begins at S_0 and goes from vertex to vertex using the sequence of arrows labeled by the letters in W.

(Refer Slide Time: 13:53)



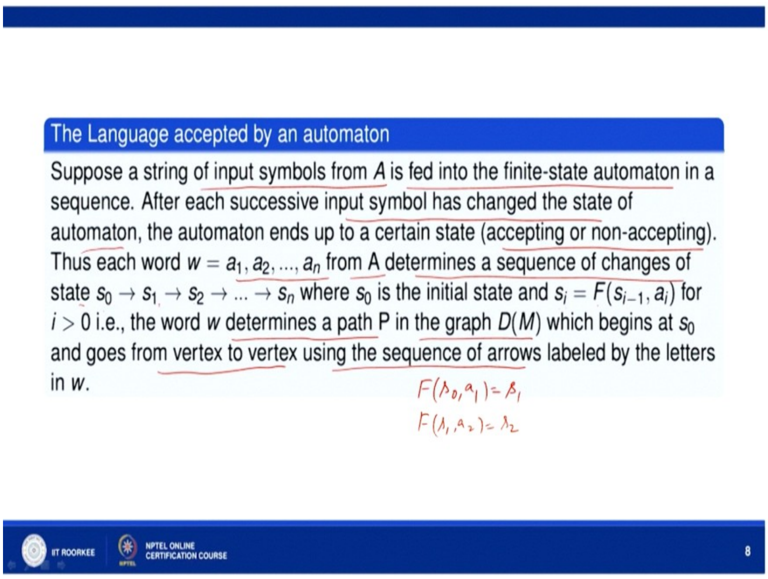


Now, let see how we get this, suppose the status of this final state, now after the sequence the final states, suppose the final states is S_n, then the state is of the final states determines whether this string is accepted by the finite state automaton, if $S_n \in$Y okay, then we say that the string is accepted by the automaton, if it does not $\in$Y we say that it is not accepted by the automaton.

These strings that send the automaton to an accepting state are called accepted or language recognised by the automaton, if I string is not accepted, we it is said to be rejected. The language L(M) of M is the collection of all words from W, The language L(M) of M is the collection of all words from A, which are accepted by the automaton M, so let us say M be an automaton with input A.

The language accepted by a finite state automaton is termed as a regular language okay, is the language accepted by a, we have seen regular language type 3 grammer is called as a regular language, so the language accepted by a language generated by type 3 grammer is called as a regular language, so the language accepted by a finite state automaton is termed as a regular language.

(Refer Slide Time: 15:18)



The status of this final state determines whether the string is accepted by the finite state automaton. These strings that send the automaton to an accepting state are called accepted or language recognized by the automaton. If a string is not accepted, it is said to be rejected. The language $L(M)$ of M is the collection of all words from A which are accepted by M.
Let M be an automaton with input A. The language accepted by a finite state automaton is termed as a regular language.

Example 2

consider the automaton M in Fig. 1. Determine whether or not M accepts the word w where

1. $w = ababba$
2. $w = baab$
3. $w = \lambda$.

$w = ababba$
$s_0 \xrightarrow{a} s_0 \xrightarrow{b} s_1 \xrightarrow{a} s_0 \xrightarrow{b} s_1 \xrightarrow{b} s_2 \xrightarrow{a} s_2$

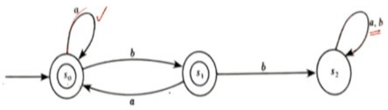
$w = baab$
$s_0 \xrightarrow{b} s_1 \xrightarrow{a} s_0 \xrightarrow{a} s_0 \xrightarrow{b} s_1$
Since $s_1 \in Y$, $w = baab \in L(M)$

$w = \lambda$
$s_0 \xrightarrow{\lambda} s_0$, since $s_0 \in Y$, $w = \lambda \in L(M)$

$F(s_0, a) = s_0$
$F(s_0, b) = s_1$
$F(s_1, a) = s_0$
$F(s_1, b) =$

$Y = \{s_0, s_1\}$
Since $s_2 \notin Y$
$w \notin L(M)$

$F(s_0, \lambda) = s_0$



State diagram of M.

Fig. 1

$F(s_2, a) = s_2$
$F(s_2, b) = s_2$

$F(s_0, a) = s_0$
$F(s_0, b) = s_1$
$F(s_1, a) = s_0$
$F(s_1, b) = s_2$

Now, let us look at this example, consider the automaton M in figure 1. Okay, automaton M in figure 1, let us look at this one, this one with this automaton okay, now here determine whether or not M accepts the word W where W is =ababba okay, so W is ababba okay, let us look at this, so now what we are saying, we begin with S_0 . Okay, we begin with S_0 and then $S_0 \rightarrow S_1$, $S_1 \rightarrow S_2$ okay, provided F ¿¿ a_1) =S_1 okay.

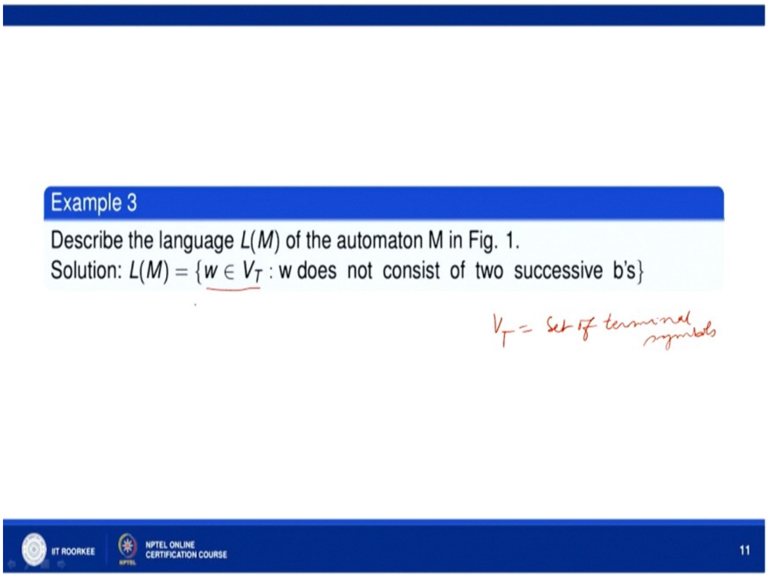
So, let see here we begin with S_0 . Okay, now the first alphabet here is letter is A, so S_0 A, so what is F ¿¿ ,a)? Let us see F ¿¿ ,a)=S_0 . Okay, so we get here S_0 , then second one is B then we need F ¿,b), let us see what is F (S_0 ,b)? Yes F (S_0,B), S_0 ,b, F (S_0, b) =S_1, so we go to S_1 okay, then third one is A, so we now need F ¿,a), F ¿,a), F ¿,a) is S_0 , then we have b okay, this one, so F ¿¿ ,b), F ¿¿ ,b) =S_1, so we have S_1 and then we have again b, so we need F ¿¿, b), F (S_1, b), yes F ¿ b) = S_2 okay.

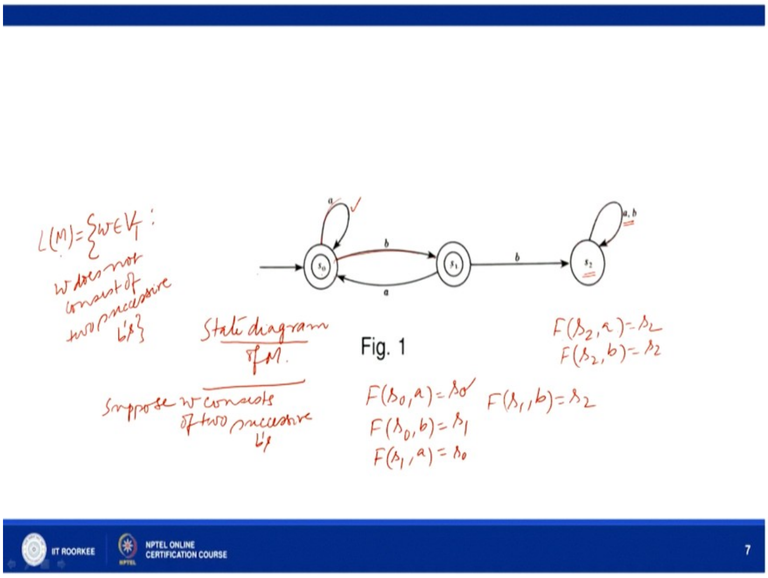
Now we have to come to A okay, so F ¿,a), F ¿,a) = S_2 okay, now what is our set Y? Y is the set of S_0 and S_1 okay, $S_2 \notin$ Y y={S_0, S_1} an element of Y okay, so since $S_2 \notin$Y okay, M do*es* not accept the word W =ababba okay, so W ∉ language determine by M. Okay, L(M), this L(M) language determine by M, L(M) is the collection of all words from A which are accepted by M and what are the words accepted by M, if the string, send the automaton to an accepting state okay, then we say that, that word ∈ L(M), so here the string, this string does not send S_0 to an accepting state, it sends to S_2 okay, which is *not* there in Y okay, so W ∉ L(M).

Now, let us look at the second case W =baab okay, so again we start with a S_0 , then S_0 b, F(S_0,b) =S_1 okay and then we have, so we have taken care of this. Now that second one is a, F(S_1, a), F ¿¿,a) is S_0 and then F ¿ , a) okay, now F ¿ , a) is what? F¿ , a) is S_0 and then we have taken the care of this, now this B, so F ¿ , b), F¿,b) is S_1 okay, so this string baab okay is there in L(M) because S_1 belongs to Y, so since S_1 belongs to Y okay, W = baab okay, ∈L(M).

Now, the last one, here third one W =λ okay, so again we started with a S_0 , S_0 λ, yes, F ¿ , λ),, F ¿ , λ), is S_0 . Okay, so F ¿ , λ), is S_0 , so the final state, this final state okay, final state is the initial state okay, since the final state is the initial state okay and S_0 is an element of Y, so since $S_0 \in$Y okay, Y is the set of accepting states okay, so it follows that W =λ belongs to L(M) okay, now λ is empty word, so this is there in S_0 .

(Refer Slide Time: 21:00)





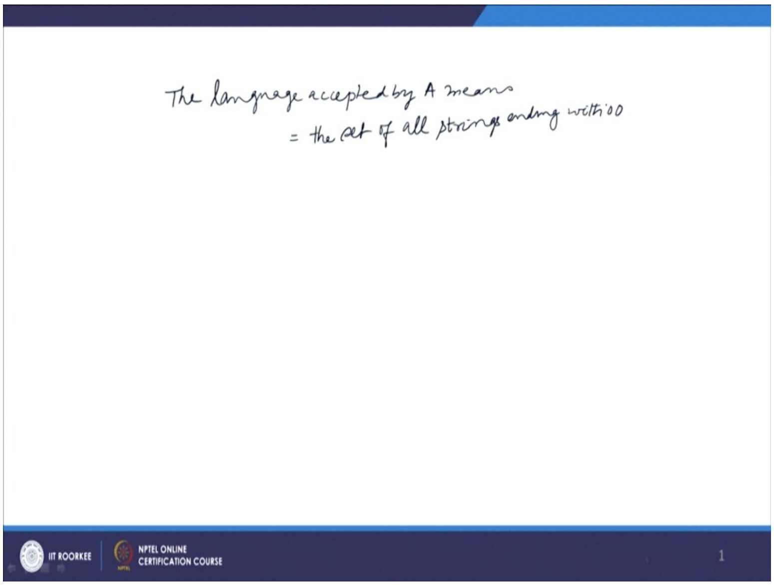
Now, let us go to another example, describe the language L(M) of the automaton M in figure 1 okay, so you can see here in this case, in this case we have to describe what is L(M)? Okay, so L(M) is the set of all W belongs, V_T is what? V_T is the set of all terminal symbols okay, so L(M) is the set of all W $\in V_T$, now in this language W should not consist of any two successive Bs why? Why we should not have two successive Bs? L(M) is the set of, so here L(M) is the set of all W, such that W does not consist of, if the word W consist of two successive Bs okay.

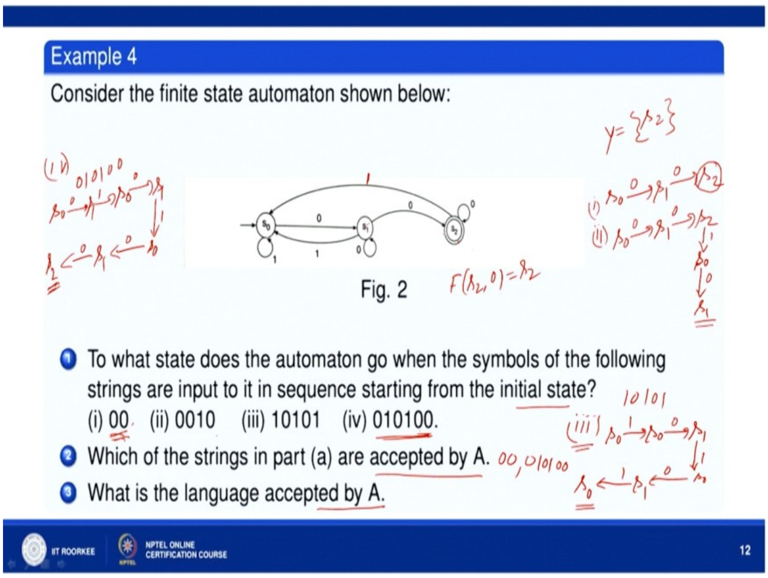
Suppose W's consist of two successive Bs then what will happen, then after two successive Bs, we will be here, in the State ¿¿ why? Because suppose before the two successive Bs we were in S_0 . Okay, then after one successive Bs we will reach to S_1, then another B will take

S_1 to ¿¿ and we will reach here in ¿¿ okay, if we were in S_1 okay, before the two successive Bs then from S_1, first from B we will take us to ¿¿ another B will take us back to ¿¿ okay.

So and we were in ¿¿ before, this two successive Bs then also okay, when B will take us back to ¿¿ and then another B will take us back to ¿¿, so we will not leave ¿¿ okay, so if W consist of two successive Bs then we will end up in ¿¿ and we will not be able to come out of S_2 and S_2 iS_0 an accepting state, accepting state are S_0 and S_1, so after two successive Bs we do not, we cannot leave S_2 and S_2 is the rejecting state and therefore this all words W, which have two successive Bs cannot be element of L(M), so the set of elements of L(M) consist of those words W, which do not have two successive Bs.

(Refer Slide Time: 24:04)





Now, let us consider this another example, we consider the finite state automaton shown below, so here there are three states S_0 , S_1 and S_2 okay, there are three states S_0 , S_1 and S_2 and there is only one accepting state as you remember accepting state is denoted by a double circle, so there is only one accepting state, so Y is a set of S_2 only, now to what state does the automaton go when the symbols of the following strings are input to it in sequence starting from the initial state.

Let see 00 okay, so we start with S_0 . Okay, then S_0 0, S_0 take 0 to S_1 okay, then next 0, so S_1 0, F ¿ 0) = S_2 okay, so the first case takes us to the from the initial state, B go to the state S_2 okay, in the case one. In the case two 0010 okay, so S_0 first 0 will take us to S_1 another 0 will take us to S_2 okay and then after that two 0, we have 1, so S_2 1, so S_2 1, S_2 1 yes, this is 1 okay, this is 1 here, so S_2 1 takes us to S_0 . Okay and then 0, so 0 okay, 0 S_0 0 goes to S_1 and S_1 is not an accepting state okay, so we just have to tell to which state we reach the starting from the initial state, if the input is 00, 0010, 10101 and so on, so in this case the state is S_2, final state is S_2, in this case the final state is S_1.

Now in the third case 10101 okay, so again start with a S_0 , S_0 1, S_0 1, yes, S_0 1 you can see here S_0 1 is S_0 . Okay, then we have 0, so S_0 takes 0 to S_1 okay and then 1 0, now 1 again, so 1, S_1 takes 1 to, S_1 takes 1 to S_0 . Okay and then 101, now we have 0, so S_0 takes 0 to S_1 okay and then we have 1, S_1 takes 1 back to S_0 , the final state is S_0 . Okay.

Now we have 010100, fourth one, 010100 let us see what is the final state here? So S_0 0 $\rightarrow S_1$, S_1 1 $\rightarrow$ S_0 . Okay, then S_0 0, 0 $\rightarrow S_1$, then 1, 1 S_1 $\rightarrow$ S_0 , then S_0 0, S_0 0 $\rightarrow S_1$, S_1 0 $\rightarrow$ S_2 okay, so here 010100 takes us to S_2, so final state is S_2. Now which of the strings in part A are accepted by A, you can see in the case one, in the case one the final state is S_2, which is an accepting state, so this string, 00 is accepted okay, so 00, this is an accepted string.

Now in second string 0010, the final state is S_1, S_1 is not an accepting state okay, so 0010 *is* accepted by a, now the third string, in the third case, the string is 10101, what is the final state? Final state is S_0 , S_0 *is not* an accepting state, so this is also not accepted by A okay, the fourth one, in the fourth case we reach S_2 okay for this string the final state is S_2, which is an accepting state, so we have this 010100 also an accepting state, also an accepted by, it is also accepted by A.

Now, what is the language accepted by A okay, now we have to tell what is the language accepted by A, so the language accepted by A, by A means the set of all W ∈A, set of all, set of all those strings which are accepted by A okay, so it is the set of all strings W, which end with two 0s okay, set of all strings ending with 00 okay, why? Because set of all words, now you can see here the first thing it has 00 okay, so it is there in the language accepted by A okay, this case there also we have two 00, it is also accepted by A.

So what we are saying is that the language accepted by A consist of all strings which end up with two 0s okay, because suppose its length is N okay, then the first N minus 2 symbols okay, will take us to anyone of those states S_0 , S_1 and S_2, the next two 0s okay, will take us to S_0 will take us to S_1, S_1 will take us to S_2, if it was there in S_0 . Okay, if it was there in S_1, then S_0 will take us to S_2 and S_2 will also take us to S_2 okay, you can see, F ¿¿ 0) =S_2 okay.

So after the N minus 2 symbols have been taken care of suppose we were in the state S_0 . Okay, then two 0s will take us to S_2 okay and if we were in the State S_1 then also two 0s will take us to S_2 and if we were in S_2, then two 0s will also take us to S_2 and S_2 is the accepting state, so we will be, if there are two 0s at the end, then it will be, that string will be accepted by the language and therefore all such strings are there in the language accepted by A, this is the, with this I would like to end my lecture. Thank you very much for your attention.